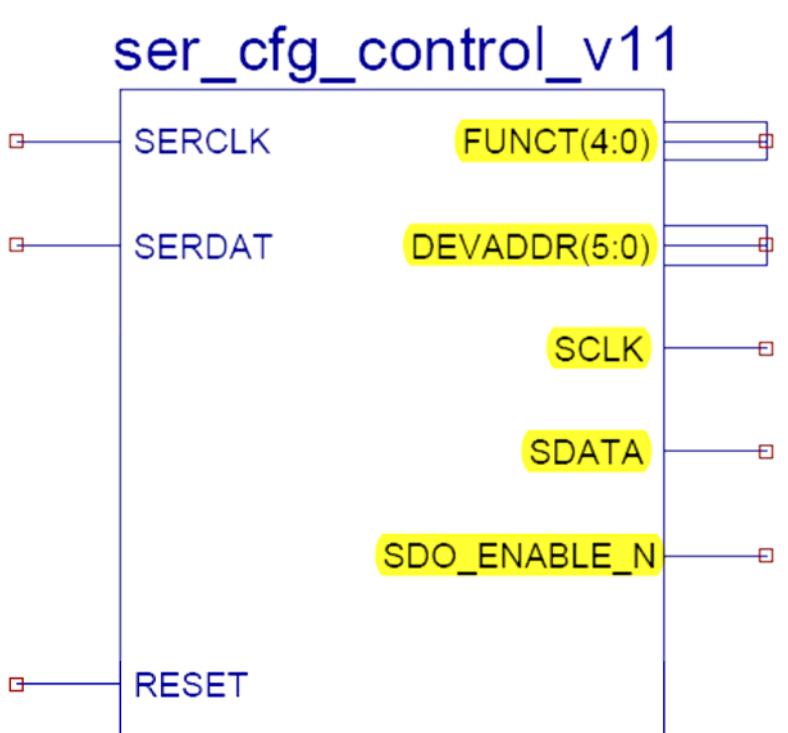


4/5/2006

-- DESCRIPTION : PROVIDES THE FRONT END FOR A COMMON, SELF DESCRIBING SERIAL  
-- DATA CONFIGURATION FUNCTION. SERDAT CONTAINS FIELDS THAT DESCRIBE THE  
-- SERIAL DATA CONFIGURATION MESSAGE LENGTH, THE CONFIGURATION FUNCTION REQUIRED,  
-- THE DEVICE ADDRESS TO CONFIGURE, AND THE ACTUAL CONFIGURATION DATA MESSAGE.  
-- THIS MODULE STRIPS OFF AND DECODES THE MESSAGE LENGTH, FUNCTION, AND DEVICE  
-- ADDRESS FIELDS AND PROVIDES THESE + THE MESSAGE ON MESSAGE SIG TO LOWER LEVEL  
-- MODULES IF REQUIRED. THE SDATA\_IN FORMAT IS:  
-- | START | FUNCTION | DEV ADDR | MSG LENGTH | MESSAGE | STOP |  
-- | 1bit | 3 bits | 6 bits | 6 bits | MSG LENGTH bits | 1bit |  
-- NOTES:  
-- SDATA IS SENT MSB FIRST.  
-- START BIT IS LOW TRUE.  
-- STOP BIT IS FALSE I.E. HIGH.  
-- MINIMUM MESSAGE LENGTH is 1 I.E. MINIMUM SERIAL STREAM LENGTH = 17 BITS.

-- IMPLEMENTATION NOTES :  
-- IMPLEMENTED FUNCTION CODES ARE:  
-- CODE 000 = ADC CONFIGURE  
-- CODE 001 = CDS CONFIGURE  
-- CODE 010 = DAC CONFIGURE  
-- CODE 011 = TEL CONFIGURE  
-- CODE 100 = AUX CONFIGURE  
--

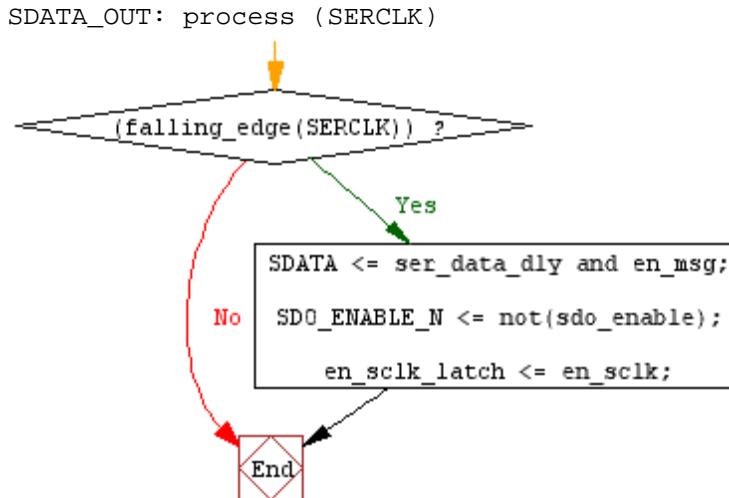


```
SERCLK      : in  std_logic;
SERDAT      : in  std_logic;
RESET       : in  std_logic;
SCLK        : out std_logic;
SDATA       : out std_logic;
SDO_ENABLE_N : out std_logic;
FUNCT       : out std_logic_vector(4 downto 0);
DEVADDR     : out std_logic_vector(5 downto 0)
```

4/5/2006

```
constant fun_len : std_logic_vector(5 downto 0) := "000010";      -- 3 bits
constant dev_len : std_logic_vector(5 downto 0) := "000101";      -- 6 bits
constant len_len : std_logic_vector(5 downto 0) := "000101";      -- 6 bits
constant TEL_WAIT_COUNT : std_logic_vector(5 downto 0) := "010011";    -- 20 clocks = 16us
constant TEL_READ_COUNT : std_logic_vector(5 downto 0) := "001100";    -- 12 bits
type CFG_STATES is (CFG_IDLE, CFG_FUNCT, CFG_DEVICE, CFG_LENGTH, CFG_MSG,
CFG_TEL_WAIT,CFG_TEL_READ);
signal cfg_state : CFG_STATES;
signal cfg_fun   : std_logic_vector(2 downto 0);
signal cfg_cntr  : std_logic_vector(5 downto 0);
signal cfg_dev   : std_logic_vector(5 downto 0);
signal msg_len   : std_logic_vector(5 downto 0);
signal en_msg     : std_logic;
signal en_sclk    : std_logic;
signal en_sclk_latch : std_logic;
signal en_function : std_logic;
signal sdo_enable : std_logic;

SCLK <= (SERCLK or not(en_sclk)) when (cfg_fun = "010") else -- DACs need sclk to remain high
when not enabled
(SERCLK and en_sclk_latch);           -- but all other functions need sclk low
DEVADDR <= cfg_dev;
FUNCT   <= "00000" when (en_function = '0') else
          "00001" when (cfg_fun = "000") else
          "00010" when (cfg_fun = "001") else
          "00100" when (cfg_fun = "010") else
          "01000" when (cfg_fun = "011") else
          "10000" when (cfg_fun = "100") else
          "00000";
```



4/5/2006

SER\_CFG : process (SERCLK, RESET)

